Year: 2012

Efficient reduction of point data sets for surface splatting using geometry and color attributes

Kim, Duck-Bong ; Pajarola, Renato ; Lee, Kwan Heng

Abstract: Surface splat, one of the point-based rendering primitives, has offered a powerful alternative to triangle meshes when it comes to the rendering of highly complex objects due to its potential for high-performance and high-quality rendering. Recently, the technological advance of 3D scanners has made it possible to acquire color as well as geometry data of highly complex objects with very high speed and accuracy. However, scanning and acquisition systems often produce surfaces that are much more dense than actually required for the intended application. Therefore, reduction of point data set is necessary to further process the model. Although many efficient sampling methods for point-based surfaces have been proposed to reduce the complexity of geometric models, none of these has taken into account color, which is fundamental for achieving a high quality visual appearance. Therefore, we propose an efficient sampling method of point data sets for surface splatting which uses both geometry and color attributes. Our proposed method converts a dense set of point samples into a sparse set of object space splats. It successfully approximates of the original model within a given geometric and color error. In order to measure color differences between point samples with consistency, the color error tolerance is evaluated in a CIELAB uniform color space.

DOI: https://doi.org/10.1007/s00170-011-3732-5

Originally published at:
Kim, Duck-Bong; Pajarola, Renato; Lee, Kwan Heng (2012). Efficient reduction of point data sets for surface splatting using geometry and color attributes. International Journal of Advanced Manufacturing and Technology, 61(5-8):787-796.
DOI: https://doi.org/10.1007/s00170-011-3732-5

Noname manuscript No.
(will be inserted by the editor)

Efficient Reduction of Point Data Sets for Surface Splatting using Geometry and Color attributes

Duck Bong Kim · Renato Pajarola · Kwan H. Lee

Received: date / Accepted: date

Abstract Surface splat, one of the point based rendering primitives, has offered a powerful alternative to triangle meshes when it comes to the rendering of highly complex objects due to its potential for high-performance and high-quality rendering. Recently the technological advance of 3D scanners has made it possible to acquire color as well as geometry data of highly complex objects with very high speed and accuracy. However, scanning and acquisition systems often produce surfaces that are much more dense than actually required for the intended application. Therefore, reduction of point data set is necessary to further process the model. Although many efficient sampling methods for point-based surfaces have been proposed to reduce the complexity of geometric models, none of these have taken into account color, which is fundamental for achieving a high quality visual appearance. Therefore, we propose an efficient sampling method of point data sets for surface splatting which uses both geometry and color attributes. Our proposed method converts a dense set of point samples into a sparse set of object space splats. It successfully approximates of the original model within a given geometric and color error. In order to measure color differences between point samples with consistency, the color error tolerance is evaluated in a $CIELAB$ uniform color space.

Duck Bong Kim · Kwan H. Lee
Gwangju Institute of Science and Technology(GIST) 1 Oryong-dong, Buk-gu, Gwangju 500-712 Republic of Korea
Tel.: +82-62-970-2423
Fax: +82-62-970-2437
E-mail: eorka333@gist.ac.kr, khlee@gist.ac.kr

Renato Pajarola
University of Zurich
Tel.: +044-635-4370
E-mail: pajarola@acm.org

Keywords point-based representation · subsampling · surface splatting · CIELAB color space

1 Introduction

Point-based representations have recently become popular due to their conceptual simplicity and flexibility [1–3]. Based on their simplicity and flexibility, 3D point data sets have motivated a variety of research on topics such as shape modeling [4], simplification [5–13], rendering [14–16], and applications [17]. The ultimate goal of point-based representations is to generate smooth and continuous surfaces from the irregularly distributed discrete point sets. Mainly there are two methods to represent geometric models using point-based representations. First, a number of researchers [1,11,18] have used moving least squares (MLS) surface models to implement geometry processing methods for point clouds. The moving least squares (MLS) surfaces provide approximating or interpolating surface for a set of point data using local polynomials. Second, the splatting surfaces [4,9,13] are represented by using the surface splat primitive which is used here to implement our proposed algorithm, see section 2.1 for more details.

The recent advancement of 3D scanners has made it possible to acquire color as well as geometry information of objects with very high speed and accuracy [19]. However, the acquired data poses a great challenge in storing, editing, transmitting, and rendering of the model due to the heavy data set. Therefore, simplification of highly detailed objects is necessary for the real time implementation and has become an important issue in many application fields such as entertainment, industrial design, human modeling, and virtual reality. For example, mobile devices, and computer games must often operate on systems where rendering and transmission capacity is highly constrained and therefore require strict control over the level of details used in models.

The main contribution of this work is to present an efficient sampling method of point data sets for surface splatting using both geometry and color attributes. It speeds up the rendering with lower sampling density compared to those of the original sampled models while maintaining the same visual quality. The proposed method generates the approximated models of point-based surfaces without maintaining the connectivity in the whole procedure within the prescribed geometric and color error. In other words, it can adapt to different sampling density of the point data sets by selecting different thresholds in both geometry and color attributes to maintain the visual quality. In addition, it is to generate the hole-free surface for surface splatting by determining the splat size based on the 3D grid method. To evaluate the proposed method, several rendering results are compared qualitatively according to the geometric and color error threshold.

2 Related work

Polygon-based simplification has been studied by many researchers such as Garland et al. [20] and Hoppe [21]. The QEM (quadric error metric) algorithm by Garland provides a fast and simple way to perform the simplification process with relatively minor storage cost. Hoppe developed a polygon simplification method in terms of a mesh optimization problem, by ordering mesh edges according to an energy minimization function. Their approaches have been refined using the appearance attributes by Garland et al. [22] and Hoppe [23], respectively. In spite of the simplicity and flexibility of polygon-based simplification, it has drawbacks in some applications. In order to simplify the meshes, each point needs connectivity or topology information, which takes a considerable amount of computation time.

The point-based sampling method, however, does not need the connectivity information nor need to preserve topology during this simplification process. Proenca et al. [5] proposed an efficient method which use the descriptive power of multi-level partition of unity (MPU) octree to generate a near optimal initial distribution of particles. Kang et al. [12] presented an efficient point-sampling algorithm from unorganized 3D point clouds using a balanced and feature-sensitive sampling algorithm based on local feature size. An efficient simplification method of point-sampled surfaces has been studied using various simplification techniques in terms of clustering, iterative simplification, and particle simulation by Pauly et al. [8]. An adaptive sampling method for MLS (moving least square) surface has been proposed using local feature size by Dey et al. [11].

Rusinkiewicz et al. [14] and Pajarola et al. [15] are to simplify splats based on hierarchical clustering schemes, which are simple and fast, however the rendering results tend to contain many redundant splats due to their hierarchical data structure. An optimized sub-sampling method for surface splats has been researched using the maximum error tolerance and global relaxation by Wu et al. [9]. Miao et al. [13] proposed a novel point-sampled geometry simplification method using an adaptive mean-shift clustering scheme. Furthermore, they developed a novel error controllable re-sampling approach for point-sampled surfaces using Gaussian sphere sampling [6]. Recently, Su et al. [7] proposed a novel curvature-aware simplification technique for point-sampled geometry based on the locally projection (LOP) operator.

Although many existing efficient sampling methods for MLS [8,11] and surface splats [9,13] have been proposed to reduce the complexity of geometric models, none of these have taken into account color, which is fundamental for achieving a high quality visual appearance. As color data is available in addition to geometry, it is desired to decimate geometric objects considering both geometry and color, since color information plays an important role in maintaining the visual quality of a point-based geometric model.

3 Overview

There are three main steps in efficient sampling of point-based surfaces. Fig. 1 and Fig. 2 illustrate the 3D point data acquisition system and the whole process of efficient sampling of point data set for surface splatting, respectively. The 3D acquisition system, the OptoScan from Breukmann [24], can acquire both geometry and color information of a 3D model several million points in a second. In order to generate a full 3D model, the model is captured multiple times in different directions. Then, 3D models from different directions are integrated into a unified 3D coordinate system using a reverse engineering technique [25]. Fig. 2 (b) shows the registered 3D model which contains 346,733 points for a duck model. Then, the dense set of point is reduced to a sparse set of object space splats, as shown in Fig. 2 (c). One of the fundamental questions in 3D geometry representation is to visualize the surface of 3D objects on a digital computer. A variety of surface representation methods, such as non-uniform rational B-splines (NURBS) and polygon representations, have been proposed in the past. However, this work investigates the idea of using point primitives for sampling of point data sets for surface splatting. Finally, the sampled point sets will be rendered by a hardware accelerated point-based rendering system [15] based on blending and visibility splatting. Fig. 2 (d) shows the rendering result using 34,512 circular splats sampled from the original data sets with the geometric error of 0.2 and the color error of 0.1.

3.1 Surface splats

Surface splats were first proposed for rendering purposes by Zwicker et al. [26]. In order to bridge the gaps among neighboring point samples, point p_i is represented with normal vector n_i and radius r_i, thereby turning it into an object-space circular disk, as shown in Fig. 3 (top). Although each surface splat contains a normal vector and a radius, using surface splat as a rendering primitive has some advantages. Surface splats

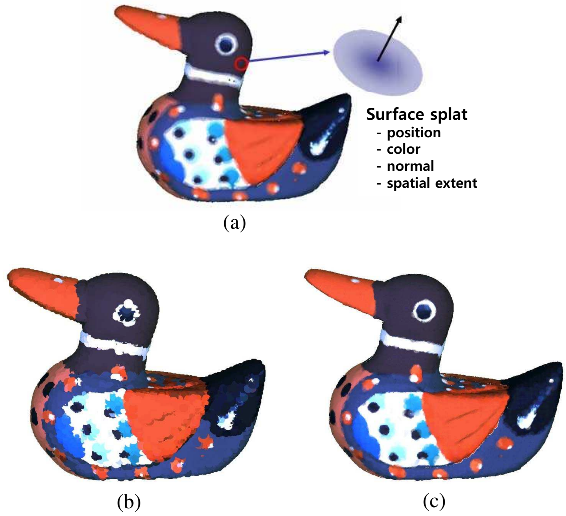

Fig. 3 Surface splat as a rendering primitive: (a) surface splat, (b) the wood duck model with 24,465 unfiltered splats, and (c) the wood duck model with 24,465 filtered splats.

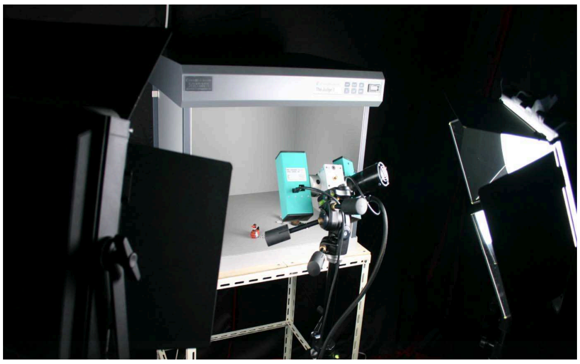

Fig. 1 3D point data acquisition system: it consists of 3D color scanner, viewing booth, and light source (D65).

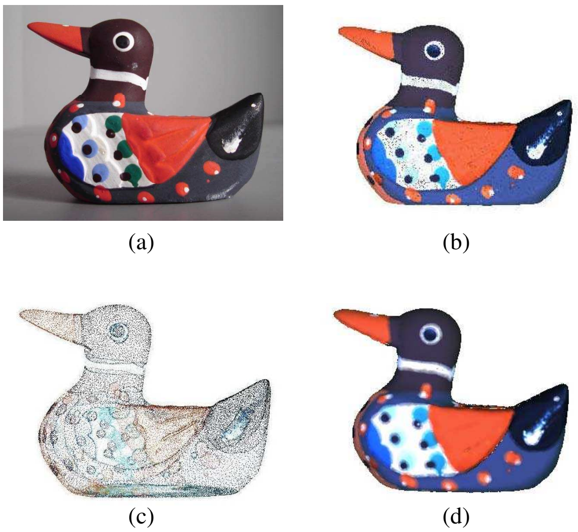

Fig. 2 Pipeline of efficient sampling of point data sets for surface splatting: (a) Real object, (b) scanned raw data, (c) subsampling, and (d) surface splatting.

do not have to be joined continuously unlike the triangles in a mesh, but are C^1 continuous instead they still provide the same topological flexibility as pure point clouds. In this research, a hardware accelerated blending and rendering system [15, 16] using circular splats with per-pixel blending and normalization is used to display the simplified point samples. The rendered image of the duck model using 24,465 filtered splats with the geometric error of 0.2 and color error of 0.1 shows better visual quality than the one using the same number of unfiltered splats, as shown in Fig. 3 (bottom).

4 Efficient sampling for surface splatting

The 3D geometric point data which contains color attribute is to be sampled efficiently according to the required level. In this research, a 3D Grid algorithm [27] is adopted to reduce the complexity of the geometric model. It subdivides the point data set into a number of sub-grids according to the prescribed geometric and color error, and each sub-grid is represented by one point. Using the 3D grid algorithm, more points are sampled in regions having high curvature and big difference in colors, as shown in Fig. 7 and Fig. 8.

A data structure based on octree has been proposed for representing geometric objects. The octree representation places the object of interest in a parallelepiped, typically a cube, which totally encloses it. As shown in Fig. 4 (a), this parallelepiped is subdivided into its 8 octants, which are then recursively subdivided a number of times based on the criteria defined by the application. The octree is used here for data reduction. The one-to-eight subdivision is determined by the standard devia-

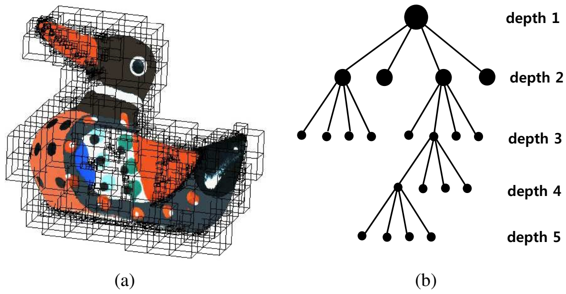

Fig. 4 3D grid generation using octree: (a) Subdivision and (b) octree in 2D case.

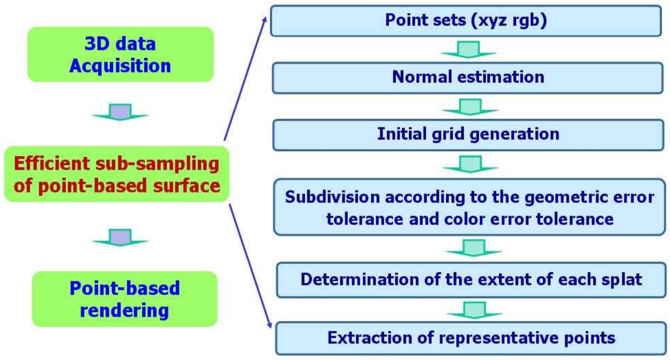

Fig. 5 Overall process of efficient sampling of point-based surface.

tion of point normal and color value compared with the given geometric and the color error in each grid.

In order to perform subdivision using the given geometric error, each normal of a point is obtained using covariance over k-nearest neighbors. Determining a radius of each splat in object space is estimated by using the length of the difference between a extracted point and a corner point in a grid. Data reduction is performed by selecting one representative point and discarding other points from each grid. The whole procedure of efficient sampling for surface splatting is shown in Fig. 5, and major steps are described in the following sections.

4.1 Normal estimation

When the normal vector of a point is not available, it can be estimated by using the neighborhood of each sample point. Since there is no connectivity information among points, the local neighborhood is usually constructed using k-nearest neighbors which can be computed by using a kd-tree. The normal vector is calculated by the principal component analysis (PCA) of the covariance matrix of the k-nearest neighbors around a given point [28]. In fact, the eigen-vectors of the covariance form a local orthogonal frame corresponding to the PCA, and thus can easily define the normal vector easily.

4.2 Subdivision using the geometry attributes

In order to generate smooth and continuous point-based surfaces with reduced point data, the point data should be distributed uniformly. However, if the 3D grid method is used without initial grids, the point data will be extracted where the geometry changes greatly. By using the initial grids, the extracted point data can be distributed uniformly before subdivision, as shown in Fig. 7 (top). The distance between extracted points is dependent on the number of the initial grids, which affects the uniformity (e.g., valence and compactness) of the model. If the user wants the extracted points to be distributed evenly to maintain regularity, a greater number of initial grids must be used. Then each grid is subdivided using octree subdivision and empty grids are again eliminated. First, the average normal vector for each grid is calculated from the normalized point normals within each grid. Next, the standard deviation of point normals within each initial grid is calculated by Equation (1).

$$\begin{aligned}\varepsilon_{geo.} &= std_{geo.} \\ &= \sqrt{\frac{\sum_{i=0}^{n}(x_i-\bar{x})^2+\sum_{i=0}^{n}(y_i-\bar{y})^2+\sum_{i=0}^{n}(z_i-\bar{z})^2}{n}}\end{aligned} \quad (1)$$

Using the prescribed geometric error, a standard deviation of normal values, we can sample more points in regions of high curvature. Fig. 6 shows the block diagram of the subdivision step, in which 3D grids are generated using a prescribed geometric and color error. When the calculated standard deviation of a grid is larger than the given geometric error, the grid is subdivided recursively. The subdivision process continues until the divided grids meet the termination condition, which is met when the standard deviation of normal within a grid is smaller than the given geometric error. Fig. 7 (top) shows the result after subdivision using a given geometric error. Fig. 7 (bottom) shows the rendering results when the model is approximated by 7,007 filtered circular splats and 12,061 filtered circular splats, respectively. There shows smaller grids around the bill and the wing of the duck model where the geometric variation occurs greatly.

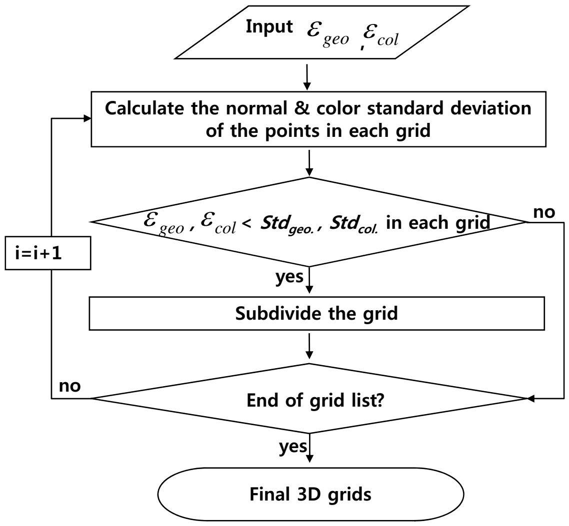

Fig. 6 The subdivision step generating 3D grids using both geometry and color variation.

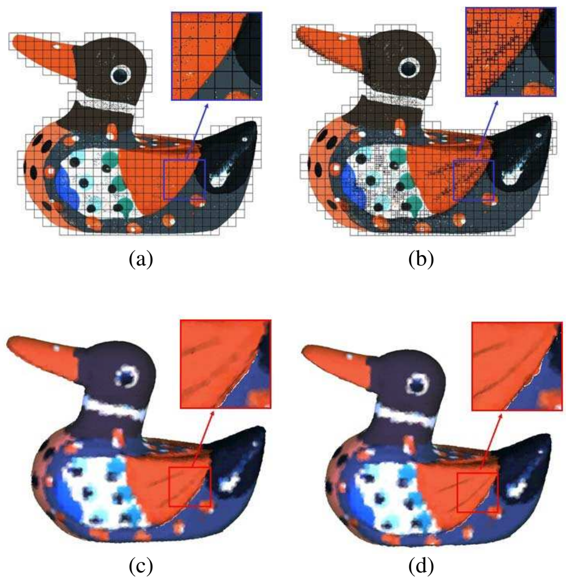

Fig. 7 The wood duck model (a) initial grids (7,007 grids) (b) subdivision using geometric error (12,061 grids) (c) rendered with 7,777 circular splat (d) rendered with 12,061 circular splats.

4.3 Subdivision using the color attributes

Although vertex colors usually come in RGB format, the RGB color space has drawbacks, since it is not perceptive-isometric. In other words, Euclidean distance between different colors in the RGB color space does not reflect the visual difference perceived by the standard colorimetric observer. For this reason, the CIELAB color space has been used to measure the color difference in many applications. Euclidean distances between different colors correspond to perceived color differences [29–31], we have also adopted this space to implement our algorithm.

The RGB values obtained from a 3D color scanner need to be converted to CIE tristimulus values to transform to the CIELAB color space as follows [31]:

$$\begin{aligned} X &= 0.4303R + 0.3416G + 0.1784B \\ Y &= 0.2219R + 0.7068G + 0.0713B \\ Z &= 0.0202R + 0.1296G + 0.9393B \end{aligned} \tag{2}$$

Then $CIEL^*a^*b^*$ coordinates are then obtained through the following non-linear transformation:

$$\begin{aligned} L^* &= 116 f(\frac{Y}{Y_0}) - 16 \\ a^* &= 500[f(\frac{X}{X_0}) - f(\frac{Y}{Y_0})] \\ b^* &= 200[f(\frac{Y}{Y_0}) - f(\frac{Z}{Z_0})] \end{aligned} \tag{3}$$

$$where$$

$$\begin{aligned} f(q) &= q^{1/3}, q > 0.008856 \\ f(q) &= 7.797q + 16/116, q \leq 0.008856 \end{aligned}$$

X_0, Y_0, and Z_0 are the tristimulus values of the illuminant, in the case of using D65 illumination. The standard deviation for using the color attributes is therefore computed in terms of CIELAB color difference units, using the following equation.

$$\begin{aligned} \varepsilon_{col.} &= std_{col.} \\ &= \sqrt{\frac{\sum_{i=0}^{n}(L_i^* - \bar{L^*})^2 + \sum_{i=0}^{n}(a_i^* - \bar{a^*})^2 + \sum_{i=0}^{n}(b_i^* - \bar{b^*})^2}{n}} \end{aligned} \tag{4}$$

Using the standard deviation of color values as the prescribed color error, we can sample more points in regions of big color difference. When the calculated standard deviation is larger than the given color error, the grid is subdivided recursively. This subdivision process continues until the divided grids meet the termination condition, which is met when the standard deviation of color within a grid is smaller than the given color error. Fig. 8 shows the rendering result after subdivision using the proposed algorithm. There are smaller grids around the eye of the wood duck model where the color difference occurs greatly, as shown in Fig. 8 (top). Fig. 8 (bottom) shows the rendering results of the model with 12,061 filtered circular splats and 24,465 filtered circular splats, respectively.

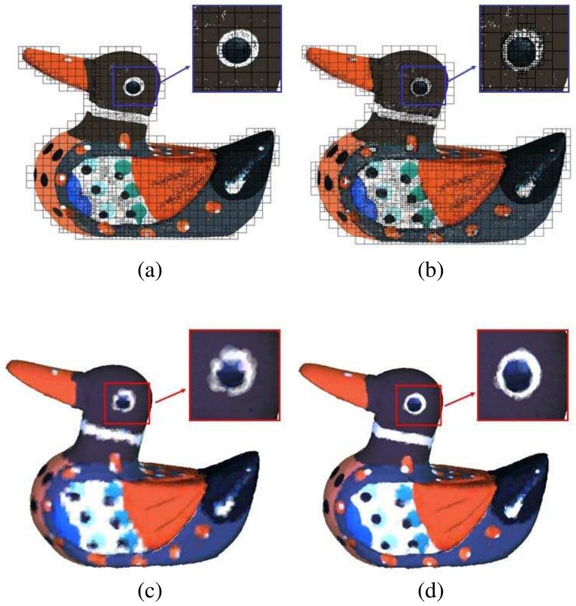

Fig. 8 The wood duck model (a) subdivision using geometric error (b) subdivision using color error (c) rendered with 12,061 circular splats (d) rendered with 24,465 circular splats.

4.4 Determining the size of each splat

After the 3D grid subdivision step, the decimated point samples are converted into surface splats in order to fill the gap between neighboring points. In this step, it is required to determine the size of each splat from the points irregularly distributed. During this process, we have to verify the current sampled set of splats actually covers the whole surface in order to guarantee a visually continuous appearance of the displayed objects. However, it is difficult to determine the size of each splat from these points due to the absence of connectivity between neighboring points. Many existing methods [8,9, 15] require extra effort to determine the size of splats to fill the gap between neighboring points. In our method, the size of each splat is determined easily using the length of the difference between an extracted point and a corner point in a grid, as shown in Fig. 9 (c).

Ideally if the extracted point is center of a grid, the size of a splat is same to the length of $|\overrightarrow{d}|$. It can guarantee the surface because each splat can enclose each grid entirely, as shown in Fig. 9 (a). However, the extracted point is not the center of a grid, meaning that we can not guarantee the whole surface with the length of $|\overrightarrow{d}|$, as shown in Fig. 9 (b). Therefore, an offset vector $\overrightarrow{\delta}$, the difference between the extracted point and the corner point, is used in the proposed method to avoid holes between surfaces splats. Using the length of radius vector $\overrightarrow{r}$, the point-based surfaces should be enclosed

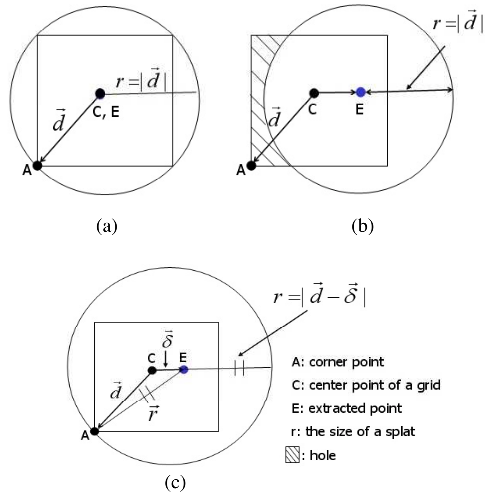

Fig. 9 Determination of the size of each splat with guarantees.

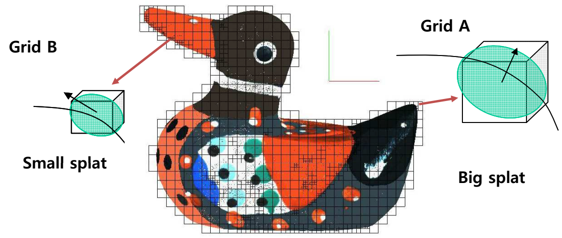

Fig. 10 The size of each splat according to the size of each grid.

by the surface splats with guarantees, as shown in Fig. 9 (c).

Sequentially when the normal and color variation changes greatly in a region, the size of a grid becomes smaller. Therefore, the size of a splat becomes smaller accordingly and vice versa, as shown in Fig. 10.

4.5 Extraction of representative points

As a result of using initial grids and grid subdivision, a number of grids are generated when the normal and color variation occurs greatly. The sample points can preserve both the geometry and color features. For selecting a representative point in a grid, the nearest point to the center of a grid is used.

5 Results and discussion

5.1 Rendering results

Our sampling and surface splatting method is applied to the wood duck and the GIST Ari models which are obtained by a 3D color laser scanner. These models are simplified from the original point set of 346,733 and

1,335,436 to a number of approximation models based on the given geometric and color error. The algorithm is written by OpenGL, and it is implemented using a 3.0 GHz Pentium 4 with a GeForceFX 6800 graphics card, running Windows.

Fig. 11 shows the visual comparison of the wood duck model using different geometric errors with the fixed color error. The original wood duck model has 346,733 points, which is approximated using circular splats by setting the geometric error with 0.3, 0.2, 0.1, and 0.08 respectively and the fixed color error. The rendering result demonstrates that geometric features look sharper as the given geometric error decreases. However, the approximated models cannot preserve the color features as the geometric error decreases.

Fig. 12 on the other hand illustrates the visual comparison of the wood duck model using different color errors with the fixed geometric error. Again, the original model is approximated using circular splats by setting the color error of 1.0, 0.15, 0.1, and 0.05 respectively with the geometric error fixed as 0.2. The rendering result illustrates that color features look sharper as the given color error decreases.

Fig. 13 shows the visual comparison between the original model (a), the model simplified by using uniform sampling method (b), the model simplified by using geometric error (c), and the model simplified by both geometric and color error (d). The simplified model has reduced the data size down to five percent of the original model. The rendering result demonstrates that the eye of the wood duck simplified by using both geometry and color attributes becomes sharp, while the result simplified by using only geometric attribute, is blurred when the same number of points are used.

Fig. 14 shows GIST Ari model using different color errors with the geometric error fixed. The original GIST Ari model contains 1.3 million points and it is approximated by circular splats using the color error of 0.3 and 0.08 with the geometric error fixed as 0.15, respectively. The rendering result also shows that color features look sharper as the given color error decreases.

5.2 Discussion

A trade-off exists between visual quality and the size of extracted surface splats. For example, it is easy to generate point-based surfaces without holes with covering surfaces using big splats, but this leads to poor visual quality due to its interferences between splats. In contrast, covering surfaces with small splats can minimize the interference between splats, but this leads to visual artifact such as holes. Considering these requirements, our proposed algorithm is to adaptively sample the point samples for surface generation without surface holes based on geometric and color error tolerance as explained in Section 4.4. In addition, a trade-off exists between visual quality and the rendering speed. The visual quality is better in accordance with the use of more surface splats extracted by the normal and color variations as explained in Section 5.1. But, the rendering speed is lower accordance with the increase of the number of surface splats.

Many previous research works [5–13] focus on proposing adaptive sampling methods only considering geometric error tolerance. Consequently, it is obvious that the geometric error from the previous methods gives better results than those from the proposed methods for its geometric error comparison. On the other hand, the proposed method deals with point sampling using geometric and color attribute with consideration of smooth surface generation. In addition, it should guarantee the hole-free surface and minimize the interference between splats. Therefore, comparing the proposed method with the previous methods is difficult, since each method for point sampling has its different specific purposes. Moreover, it is difficult to compare the proposed method in a quantitative manner, since we consider the geometric as well as color error. Therefore, we compare the rendering results visually as shown in Fig. 11, 12, 13, and 14.

It is concluded that the proposed method adaptively decimates the original 3D model with feature-preservation of geometric and color features as shown in Fig. 11, 12, 13, and 14. In addition, it guarantees the hole-free surface by enclosing each grid entirely based on an offset vector. Based on the flexibility and simplicity of the point-based representation, the proposed method can be widely used, since many 3D geometric models with color attributes are widely applied in many applications (e.g., game, movie, etc). However, it is difficult to handle the point clouds with large noises or highly non-uniform distributed samples. To overcome this limitation, some post-processing procedure (e.g., de-noising, smoothing, etc) from the 3D scanned surface data can be performed for its re-sampling task.

6 Conclusion

An efficient sampling method of point data for surfaces splatting is proposed using both geometry and color attributes without using any connectivity information. It can generate a smooth surface with guaranteeing the hole-free surface and minimizing the interference between splats by enclosing each grid entirely. In other words, the number of point samples is determined in

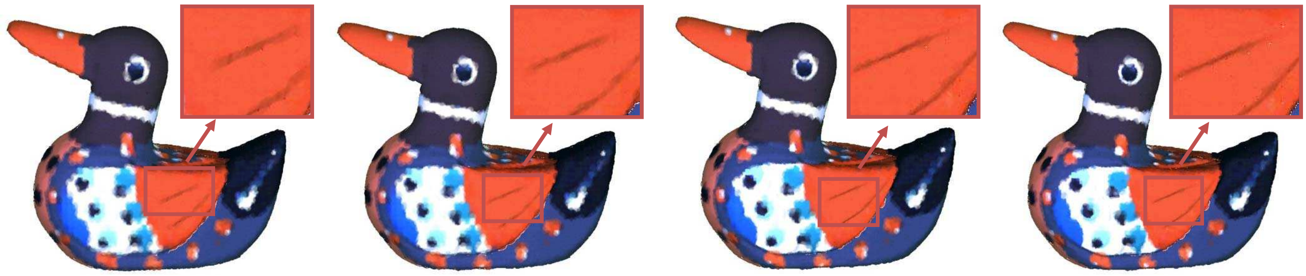

Fig. 11 Comparison of wood duck models using different geometric error with the same color error: From left to right shows the model with 8,600, 12,377, 31,575, and 43,754 filtered circular splats and the geometric error of 0.3, 0.2, 0.1, and 0.08 respectively with the same color error of 0.2.

Fig. 12 Comparison of wood duck models using different color error with the same geometric error: From left to right shows the models with 12,061, 15,630, 24,465, and 42,952 filtered circular splats and the color error of 1.0, 0.15, 0.1, and 0.05 respectively with the same geometric error of 0.2.

Fig. 13 Comparison of wood duck models using the same sampling density: (a) original point samples 346,733, (b) 16,827 filtered circular splats simplified by uniform sampling, (c) 16,913 filtered circular splats simplified by using only geometric error of 0.152, (d) 16,800 filtered circular splats simplified by using geometric error of 0.2 and color error of 0.14.

Fig. 14 GIST Ari model (a) the original model with 1,335,436 points, (b) the model with 64,758 filtered circular splats, (c) the model with 89,139 filtered circular splats.

accordance with the geometric and color error tolerance. The extracted sampled points are non-uniformly distributed and can account for the local geometric and color features. For example, the sampled points are dense in the high normal and color variation regions, while they are sparse in the low normal and color variation regions. In addition, the size of spats in the regions where the normal and color variation changes greatly becomes smaller accordingly and vice versa. Experimental results demonstrate that the visual quality is better as the given prescribed geometric or color error decreases. The model simplified by using both geometry and color attributes show the best quality compared with other conditions when the resulting models have the same number of points.

In future we will define new mathematical criteria for determining the optimum size of each splat. It will guarantee the use of local feature sizes in generating point-based surfaces. We will also research a refinement algorithm to generate better point-based surfaces from the irregularly distributed point samples.

Acknowledgements This work was supported by the National Research Foundation (NRF) grant funded by the Korean government (MEST) (No. 20100018897).

References

1. Gross M, Pfister H, Alexa M, Dachsbacher C, Pauly M, Baar JV, Stamminger M, Zwicher M (2003) Point-based computer graphics. In Eurographics 2003 Tutorial T1
2. Kobbelt L, Botsch M (2004) A survey of point-based techniques in computer graphics. Computers and Graphics 28(6): 801-814
3. Gross M (2009) Point based graphics: state of the art and recent advances. In ACM SIGGRAPH 2009 Courses (SIGGRAPH '09). ACM, New York, NY, USA, Article 18, 68 pages
4. Pauly M, Keiser R, Kobbelt L, Gross M (2003) Shape modeling with point-sampled geometry. In Proceedings ACM SIGGRAPH 2003, pp 641-650
5. Proenca J, Jorge J, Sousa M C (2007) Sampling Point-Set Implicits. Proc. Eurographics Symp. on Point-Based Graphics, p.11-8
6. Miao Y, Diaz-Gutierrez P, Pajarola R, Gopi M, Feng J (2009) Shape Isophotic Error Metric Controllable Resampling for Point-Sampled Surfaces. IEEE Int. Conf. on Shape Modeling and Applications, p.28-35.
7. Su Z X, Li Z Y, Zhao Y D, Cao J J (2011) Curvature-aware simplification for point-sampled geometry. Journal of Zhejiang University SCIENCE C 12(3): 184-194
8. Pauly M, Gross M, Kobbelt LP (2002) Efficient Simplification of Point-Sampled Surfaces. Proc. IEEE Visualization conference, Boston, MA, October 2002, pp 163-170
9. Wu J, Kobbelt LP (2004) Optimized Sub-Sampling of Point Sets for Surface Splatting. Computer Graphics Forum 23(3):643-652
10. Moenning C, Dodgson NA (2003) A new point cloud simplification algorithm. 3rd IASTED International Conference on Visualization, Imaging, and Image Processing (VIIP 2003) 8-10 Sep. 2003, Benalmena, Spain, pp 1027-1033
11. Dey TK, Sun J (2005) An Adaptive MLS Surface for Reconstruction with Guarantees. Tech. Rep. OSU-CISRC-4-05-TR26, Apr. 2005.
12. Kang E, Kim D, Lee K (2008) Balanced feature-sensitive point sampling for 3D model generation. Int J Adv Manuf Technol 38(1-2):130-142
13. Miao Y, Pajarola R, Feng J (2009) Curvature-aware adaptive re-sampling for point-sampled geometry. Computer-aided Design 41(6):395-403
14. Rusinkiewicz S, Levoy M (2000) Qsplat: A Multiresolution Point Rendering System for Large Meshes. Proc. SIGGRAPH 2000, pp 343-352
15. Pajarola RB, Sainz M, Guidotiti P (2004) Confetti Object-Space Point Blending and Splatting. IEEE Trans Vis Comput Graph 10(5):598-608
16. Ren L, Pfister H, Zwicker M (2002) Object Space EWA Surface Splatting: A Hardware Accelerated Approach to High Quality Point Rendering. Computer Graphics Forum 21(3):461-470
17. Kim HJ, Bickel B, Gross M, Choi SM (2010) Subsurface scattering using splat-based diffusion in point-based rendering. Science China Information Sciences, 53(5) 911-919
18. Alexa M, Behr J, Cohen-Or D, Fleishman S, Levin D, Silva C (2001) Point set surfaces. In Proc. IEEE Visualization conference, San Diego, California, October 2001, pp 21-28
19. Levoy M, Pully K, Curless B, Rusinkiewicz S, Koller D, Pereira L, Ginzton M, Anderson S, Davis J, Ginsberg J, Shade J, Fulk D (2000) The digital Michelangelo project: 3D scanning of large statues. In Proceedings SIGGRAPH 2000, pp 131-144
20. Garland M, Heckbert PS (1997) Simplification Using Quadric Error Metrics. In Proceedings of ACM Siggraph, Los Angeles, California, August 1997, pp 206-216
21. Hoppe H (1996) Progressive Meshes. In: Proceedings of ACM Siggraph, New Orleans, August 1996, pp 99-108
22. Garland M, Heckbert PS (1998) Simplifying surfaces with color and texture using quadric error metrics. Proceedings of the IEEE Visualization 98, Los Alamitos, California, October 1998, pp 263-269
23. Hoppe H (1999) New quadric metric for simplifying meshes with appearance attributes. Proceedings of IEEE Visualization
24. Breuckmann. http://www.breuckmann.com/.
25. Varady T, Martin RR, Cox J (1997) Reverse Engineering of Geometric Models?An Introduction. Computer Aided Design 29(4):255-268
26. Zwicker M, Pfister H, Baar J, Gross M (2001) Surface Splatting. Proc. SIGGRAPH 2001, pp. 371-378
27. Lee K, Woo H, Suk T (2001) Point Data Reduction Using 3D Grids. Int J Adv Manuf Technol 18(3):201-210
28. Jolliffe I (1986) Principle Component Analysis, Spring Verlag
29. Rigiroli R, Campadelli P, Pedotti A, Borghese NA (2001) Mesh Refinement with color attributes. Computer and Graphics 25:449-461
30. Robertson PK (1988) Perceptual Color Spaces. IEEE Computer Graphics and Applications 8(5):50-64.
31. Fairchild MD (2005) Color Appearance Models, 2nd ed. John Wiley and Sons, England